HBD clusterizer with built in background subtraction

Ermias Atomssa

HBD meeting

2010.06.02

Intro 1

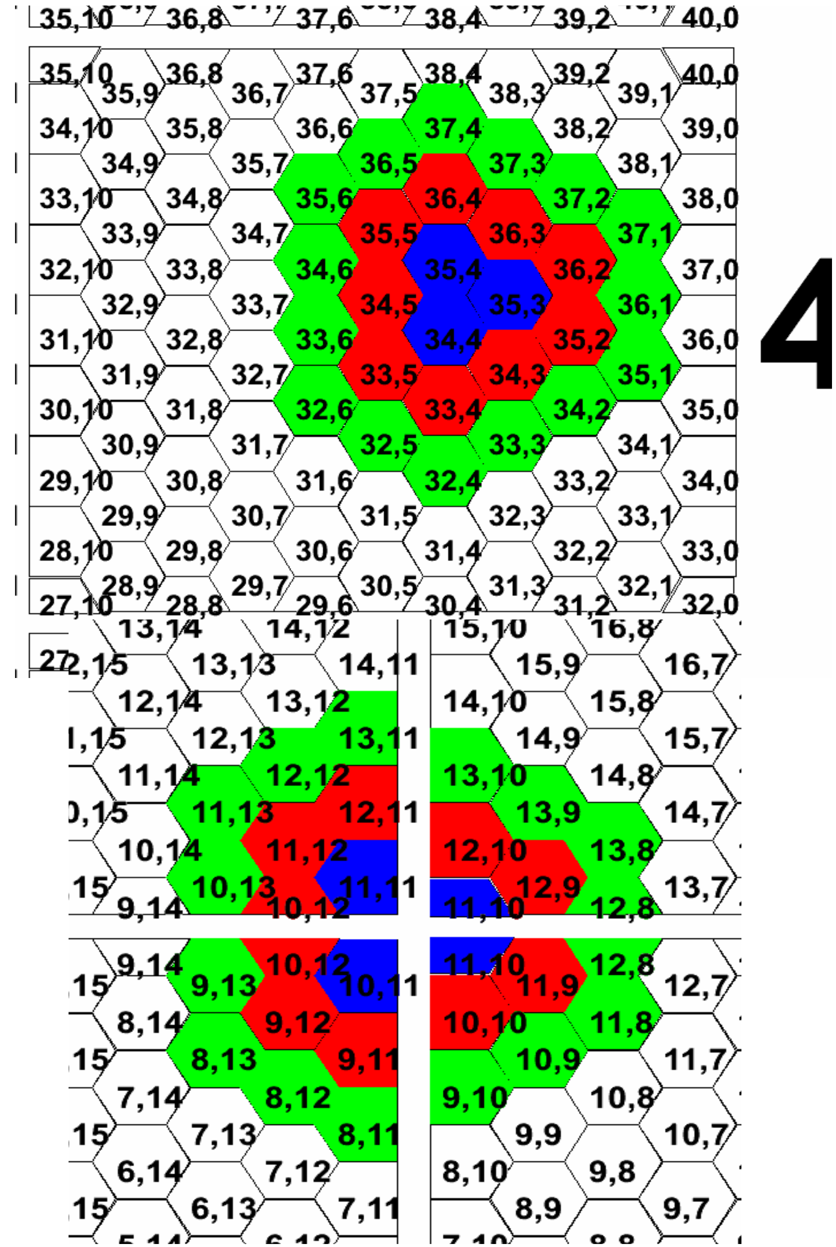
- An alternate clusterization algorithm is being developed
  - Main point: Background is handled by subtracting average per pad background estimated from surrounding area
  - It is still under development and testing but for people interested, its been submitted to cvs (in offline/analysis/hbd_proto)
    - It works like any other analysis module
    - Clusterizer: offline/analysis/hbd_proto/HbdLbsClusterizer
      - For this to work, one has to locally compile offline/packages/hbd after editing Makefile.am to add HbdBlobListv1.h to install headers
    - Embedding tests: offline/analysis/hbd_proto/HbdEmbed
    - Ntuples and plotting: offline/analysis/hbd_proto/HbdAnalysis
    - Simulation tuning: offline/analysis/hbd_proto/HbdMcChargeRecal
  - There is still a lot of debugging couts and some valgrind errors.
    - The code can be improved in efficiency and style
    - Any input is welcome, and feel free to modify if you have ideas or let me know

Intro 2: A new clusterization algorithm

- Better of the two worlds:
  - Like Weizmann clusterizer: two steps, “preclusterization” and merging.
    - But, before merging there is a control step where preclusters are selected based on a few criteria
  - Like HnS clusterizer: preclusters are triplets, most natural shape for the hexagonal symmetry of the HBD pads
    - It doesn't need to depend on the projection of electrons even in high background environment. Though this information can be used if needed.
- And a little bit more....
  - At the preclusterization step, a local background subtraction is internally (without the use of parametrization) applied.
    - This is done by estimating the background level from neighboring pads of the precluster. There seem to be (cf slide 5) reasonable correlation to warrant this
  - After merging, the final cluster's background is subtracted using neighboring pads
  - For this reason, will refer to the new clusterizer as of LBS (local background subtraction) method

Preclusterization

- First step of the algorithm is the selection of preclusters.
  - Candidates for preclusters are all possible compact triplets in the HBD (def. All members sharing a single edge with the other two members)
- Preclusters have
  - first neighbors
  - and second neighbors.
  - And they cross borders
- They have the following properties:
  - Charge & area of Members
  - Charge & area of 1st & 2nd neighbors
  - Net signal in the “member” zone
  - “Shape” meaning distribution of net charge among pads in member zone



Justification of background estimation

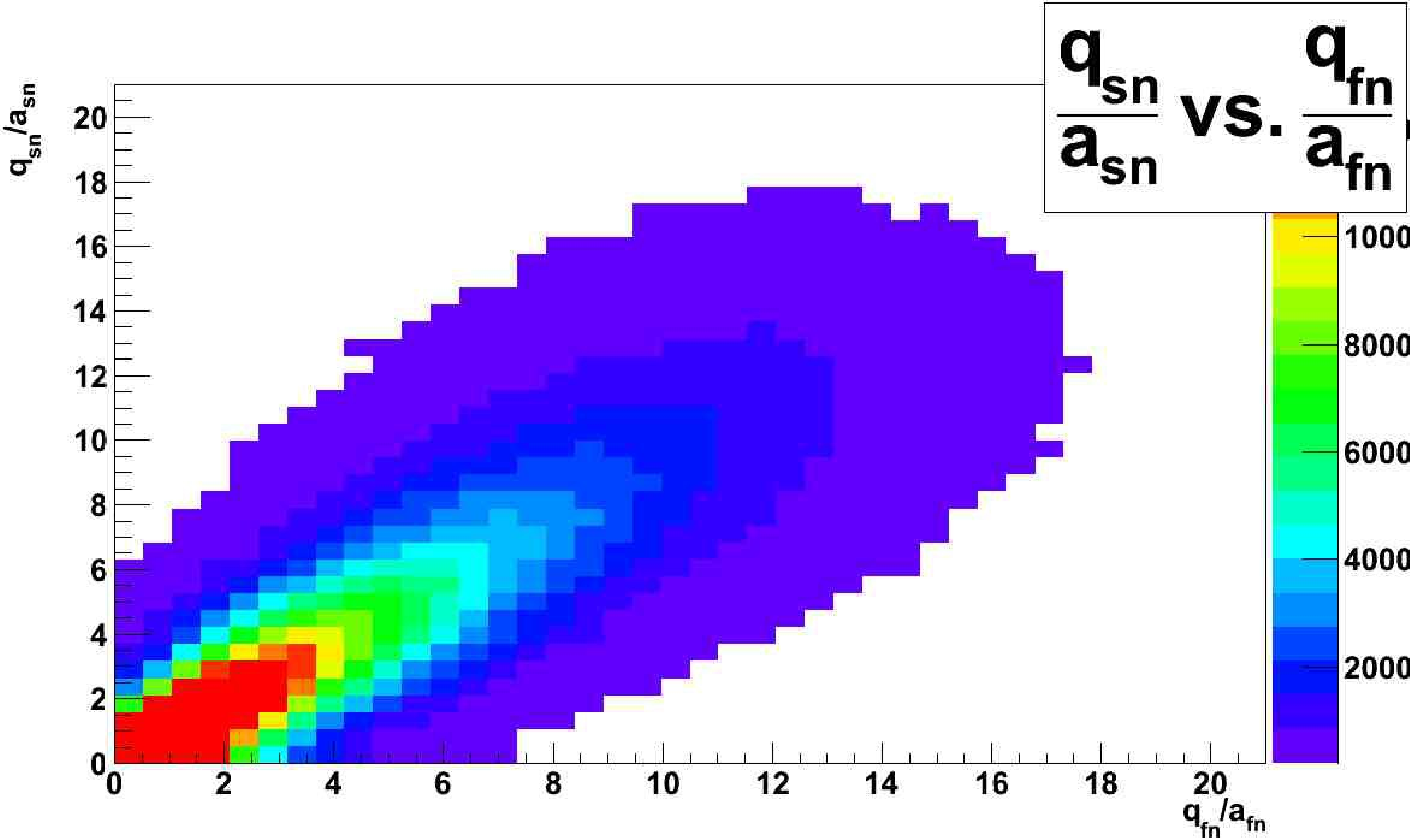
- Basic assumption of the method
  - Scintillation background varies continuously over HBD surface
  - Background in any compact group of pads can be estimated from the average rate of npe in its neighboring pads

$$bkg = a_{mem} * \left(\frac{w_{fn} * q_{fn}}{a_{fn}} + \frac{(1 - w_{fn}) * q_{fn}}{a_{sn}} \right)$$

mem=triplet member fn=first neighbor, sn=second neighbor
a=area, q=number of photoelectrons
w= weight, for now set to 0.5

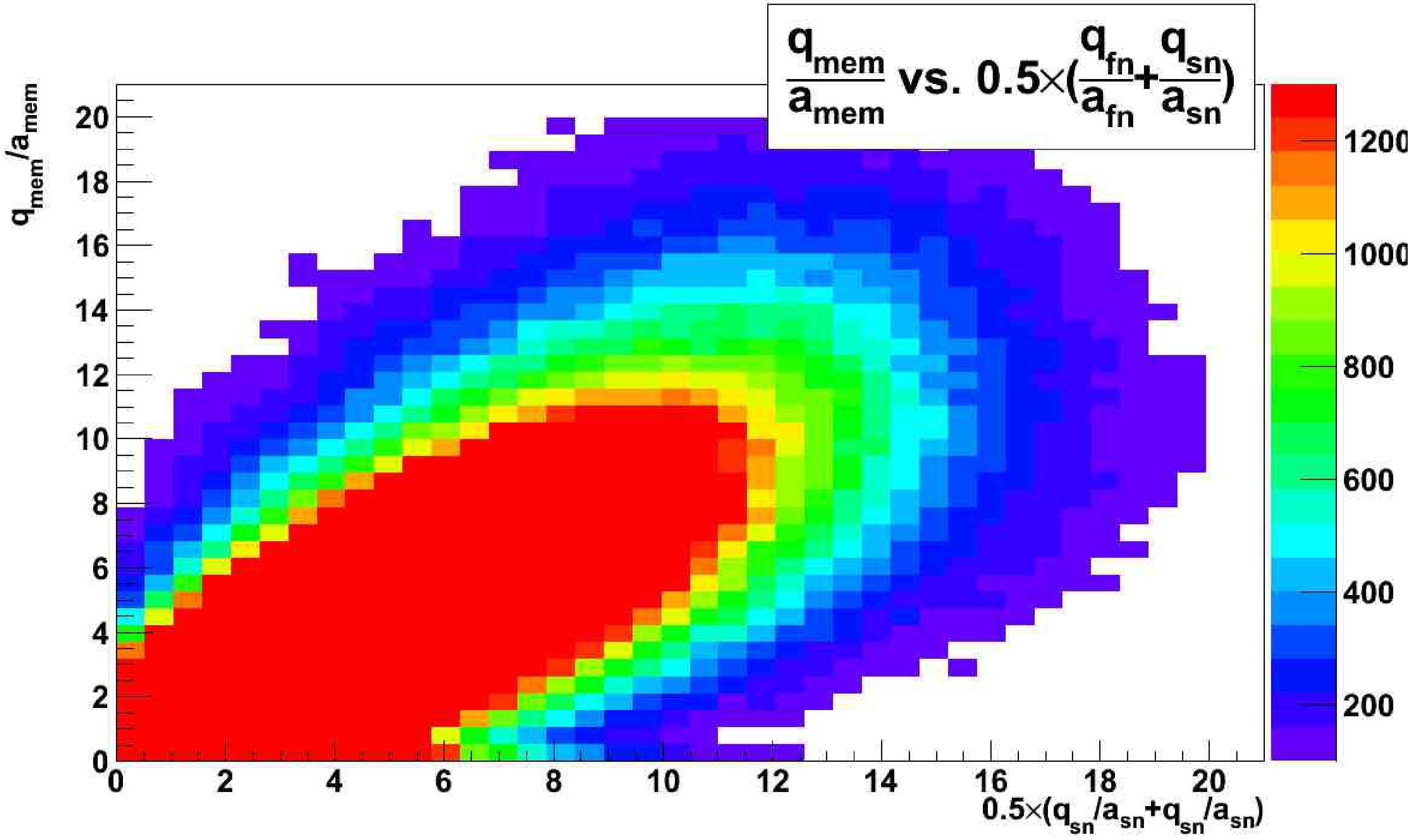
Justification of background estimation

- Basic assumption of the method
  - Scintillation background varies continuously over HBD surface
  - Background in any compact group of pads can be estimated from the average rate of npe in its neighboring pads



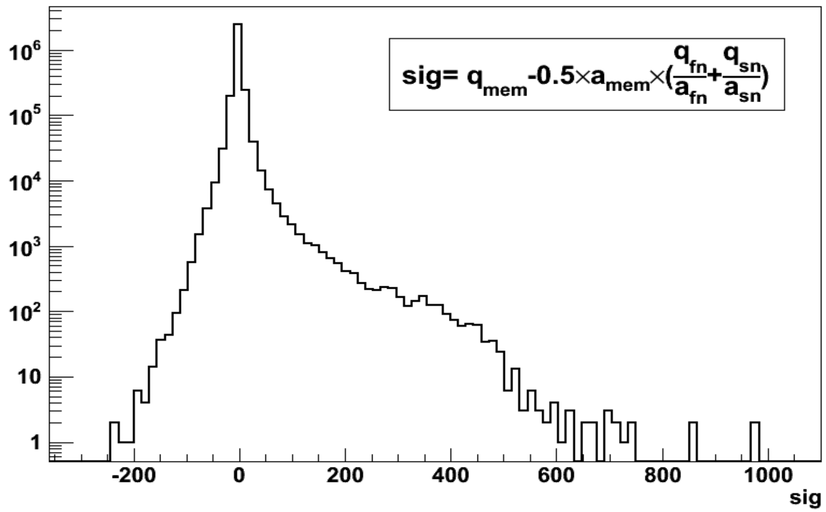
Justification of background estimation

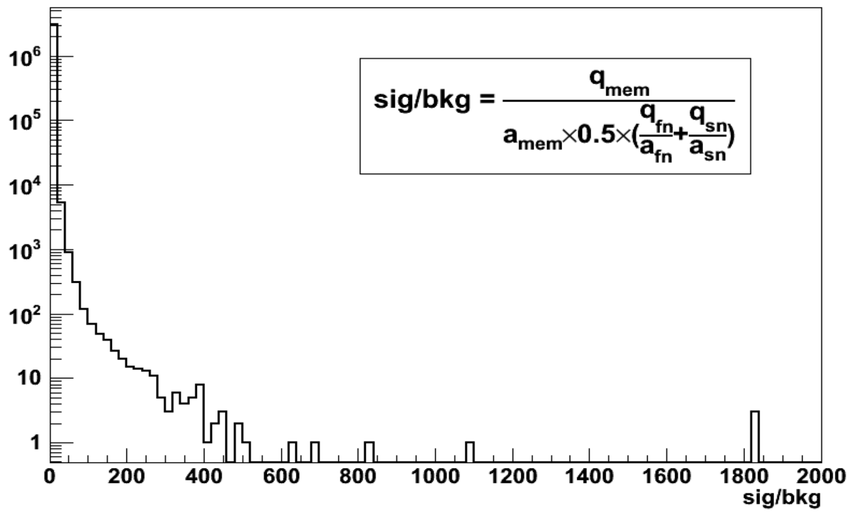
- Basic assumption of the method
  - Scintillation background varies continuously over HBD surface
  - Background in any compact group of pads can be estimated from the average rate of npe in its neighboring pads



Precluster selection

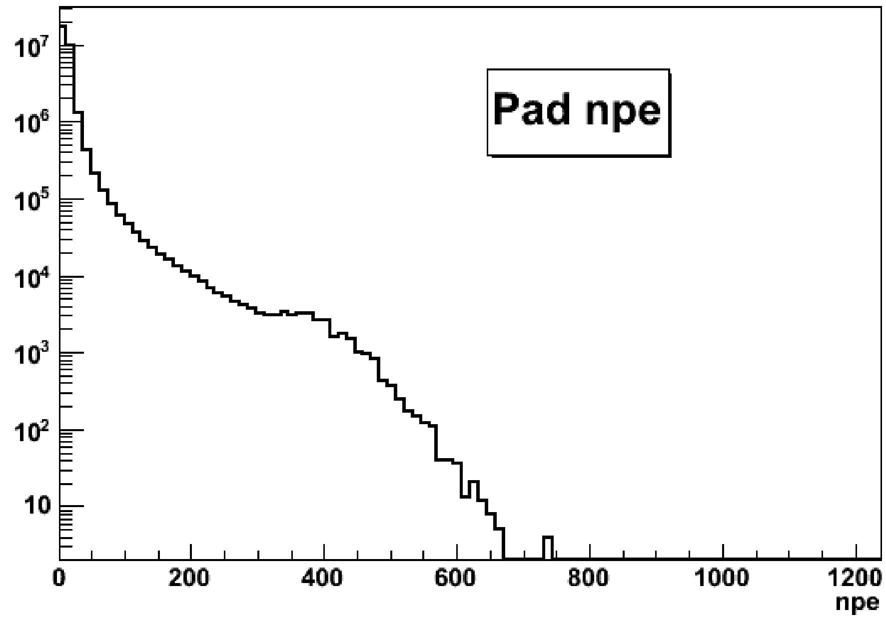
- Don't want to keep everybody
  - Code will be slow
  - Will end up with superbig clusters
- What to keep?
  - Reasonable net signal
    - For now keeping 5<sig(npe)<50
    - This spans both the singles and doubles expected charge in a triplet
  - Reasonable S/B
    - We can cut on estimated S/B
    - Optimization will be shown later
  - Shape cut
    - Distribution of a couple of such parameters will be shown later for data and MC

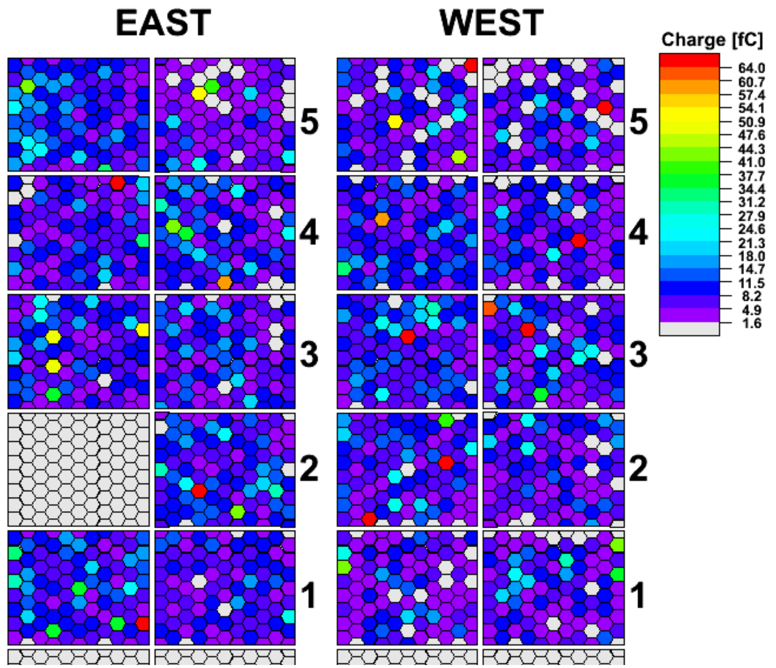




HIPs: an issue with a solution

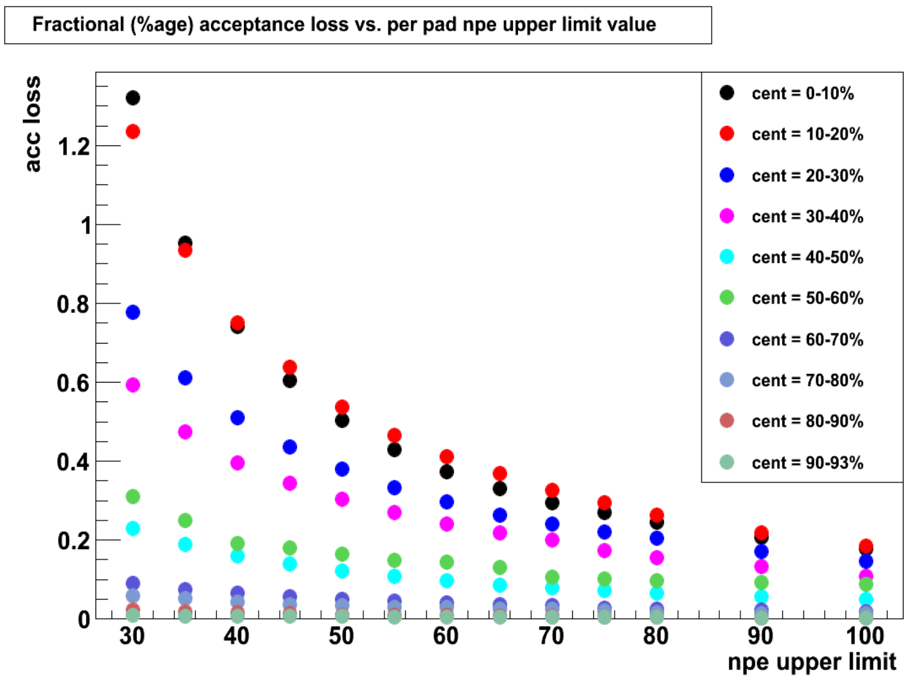
- The pad by pad charge distribution has a very long tail
  - Caused by physics processes that deposit a huge amount of energy
  - Much more than typical per pad charge expected from either scintillation or Cerenkov
  - Rate is proportional to intensity
  - X-ray, neutrons heavy particles?
- These pads if left alone are a big problem for any clusterization algorithm, because they can seed fake clusters.
- Fortunately, event by event, they cover only a very small fraction of the active HBD area

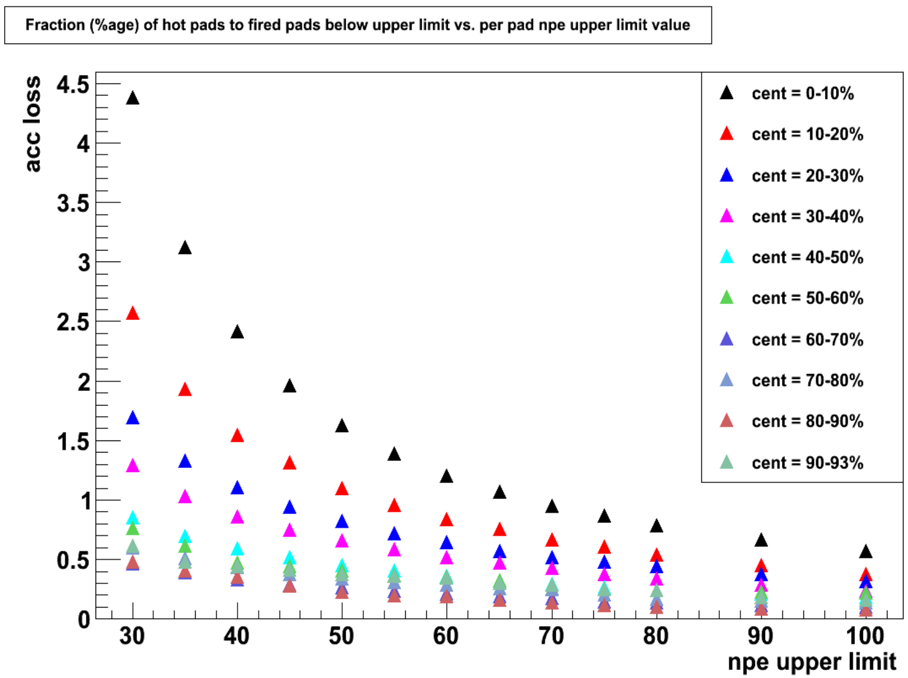




Effect of upper limit on pad npe

- Before clusterization one can set npe=0 for those pads that fire above a certain upper limit
  - Plot on left: Event averaged fraction of acceptance loss incurred by throwing out pads firing above un UL, vs. the value of the ul for different centralities
  - Plot on right: Fraction of pads firing above upper limit to those firing below upper limit but still above threshold

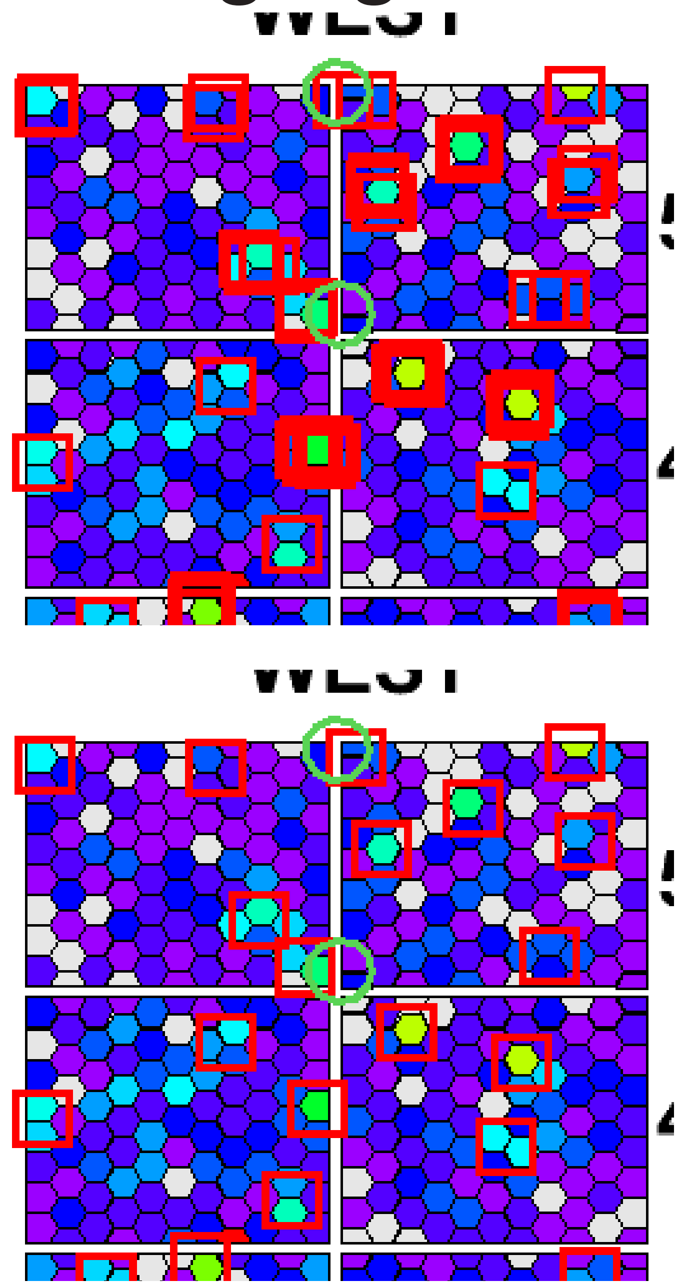




Cutting at 50 seems safe. <2% of fired pads are lost even in most central event

Merging and post merging

- Overlapping preclusters
  - Share atleast one pad
- Final clusters
  - Lump together pads from all overlapping groups of preclusters
- Local bkg. subtraction
  - Merged clusters have 1^{st} and 2^{nd} neighbors just like preclusters
  - 1^{st} and 2^{nd} neighbor charge is used to estimate background to subtract from the members of merged cluster
- Cluster track association
  - Nothing new here, based on proximity just like in Wis & HnS

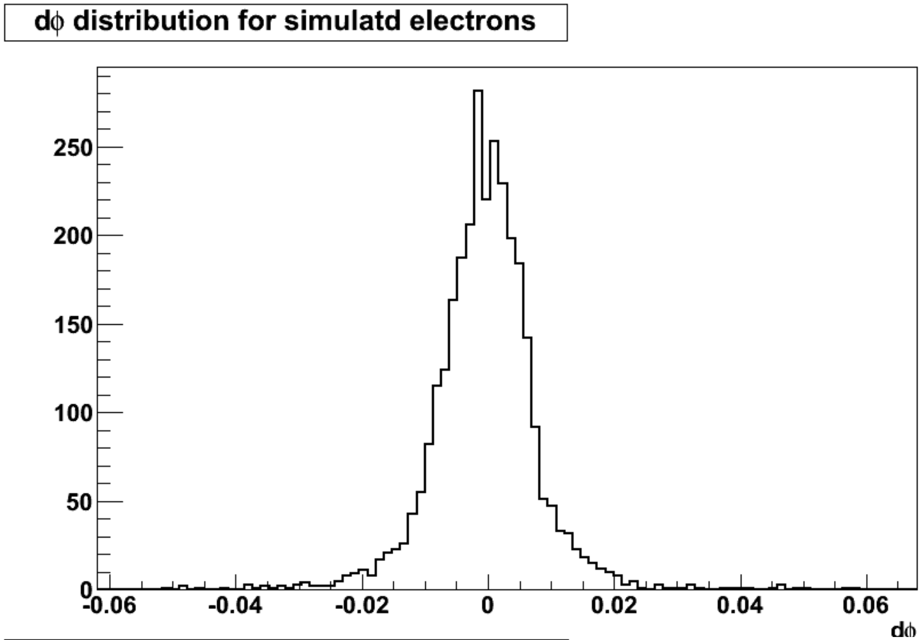


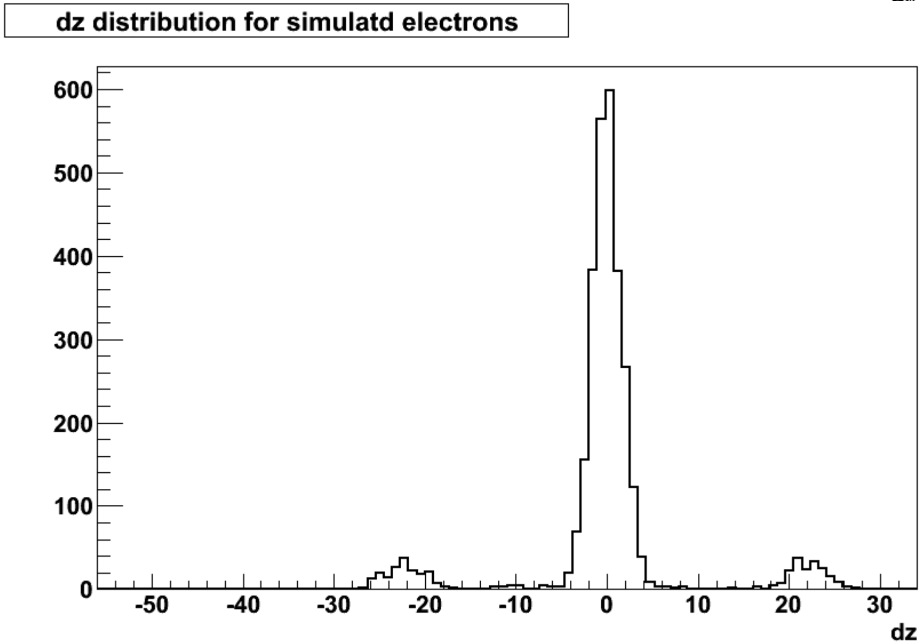
Validation

- For the validation here is the program
  - Single electron simulation with no background
    - Simulation tuning, geometry cross check, shape study
  - Single electron simulation with “emulated” background
    - Optimization of precluster selection criteria, fake rate, cluster size, cluster rates
  - Double electron simulation (Conv. and Dalitz) w/ & w/o emulated Bkg
    - Confirm doubling of the cluster signal, estimate misidentification rate from doubles created midway inside the HBD
  - p+p events
  - Event Accumulator/ Embedding
    - More realistic background. Do we still get same answer from the clusterizer for simulated electrons?
    - Embedding already implemented (initial test on single electrons)
  - Real Au+Au data
    - Cluster shape, singles/doubles/hadron charge comparison, Analysis

Single electrons, no background

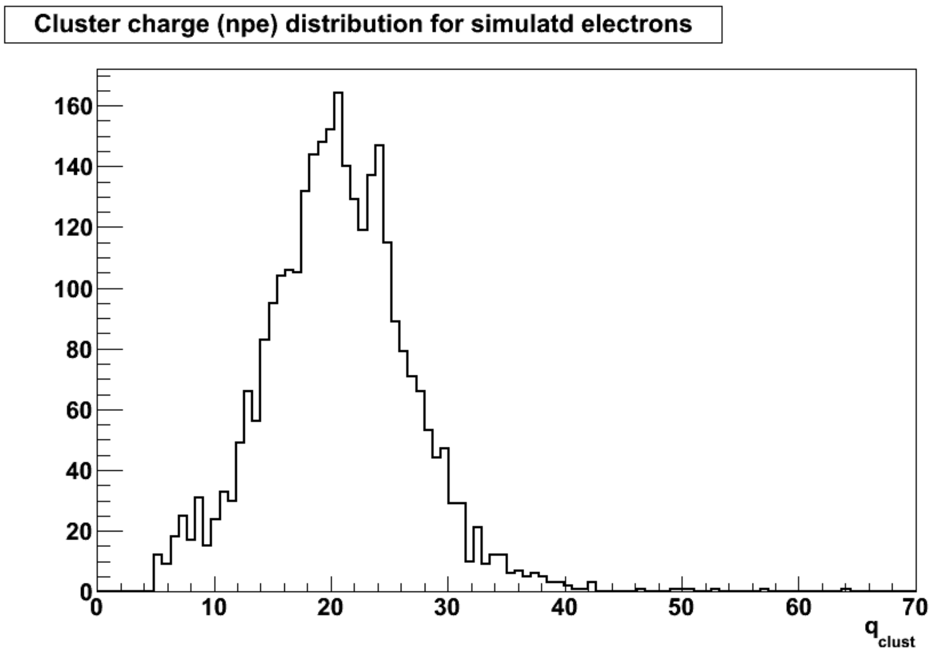
- Usual PHENIX chain
  - x,y,z = 0,0,±20
  - Full Hbd response
- Run clusterizer
  - Dphi, Dz look very good
    - Except for wings at +-20cm for dz
- This demonstrates that the geometry is being used correctly in the code.
-

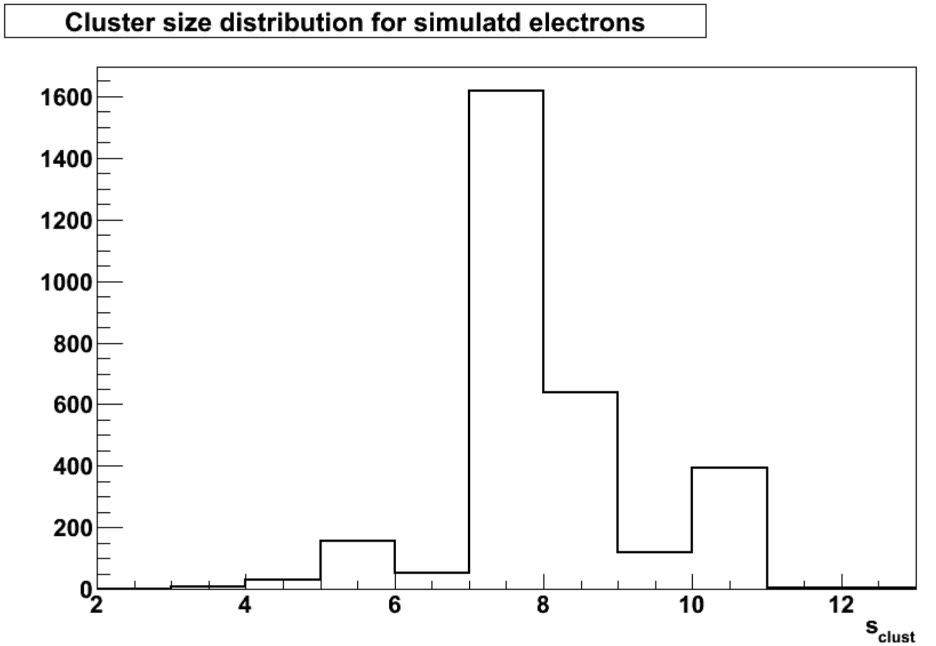




Cluster charge and size distributions

- Cluster charge distrib.
  - Off by a factor of ~2.7
    - Running a 'Recal' module that divides every pad by this factor
    - This should be done only on Cerenkov signal
- Cluster size distribution
  - Cluster sizes are somewhat big.
  - Current merging mechanism tends to add 1st neighbors
  - This should not have too much effect on the cluster charge since background is subtracted event by event



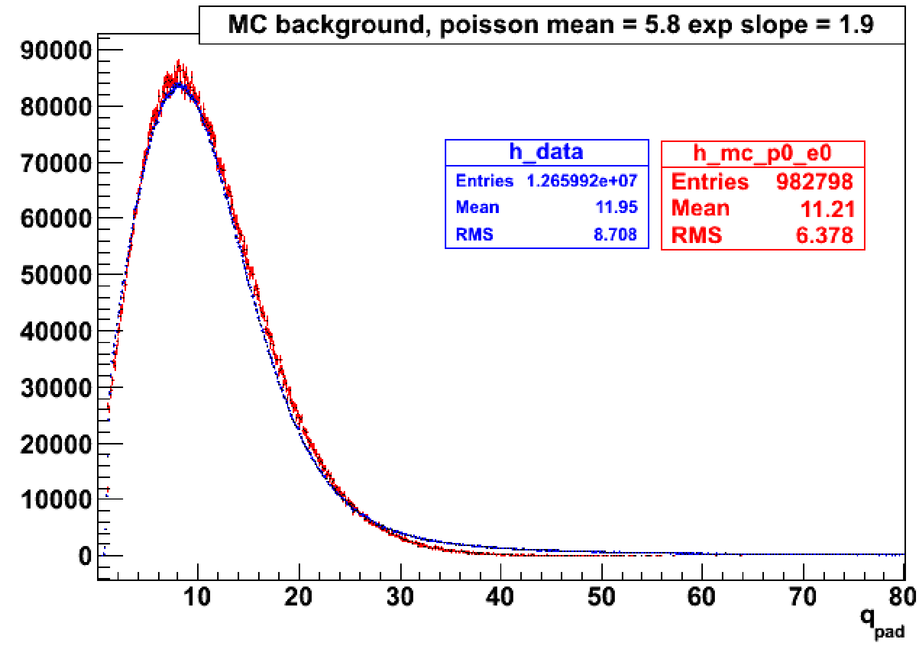


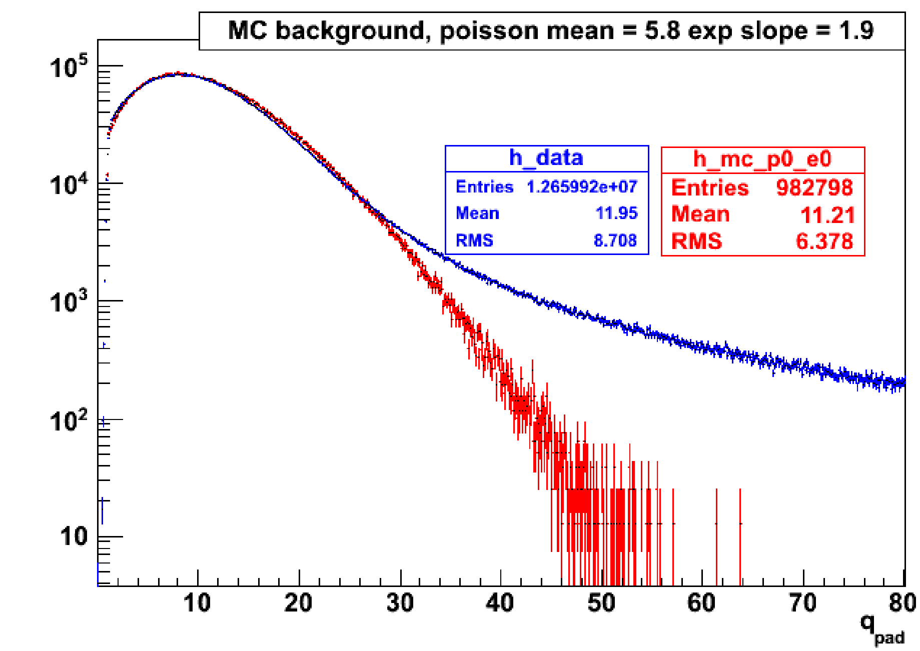
Mimic the real data backgrond

- Attempt to generate RD like background

$$q = \sum_{0}^{P(M)} \exp(\tau)$$

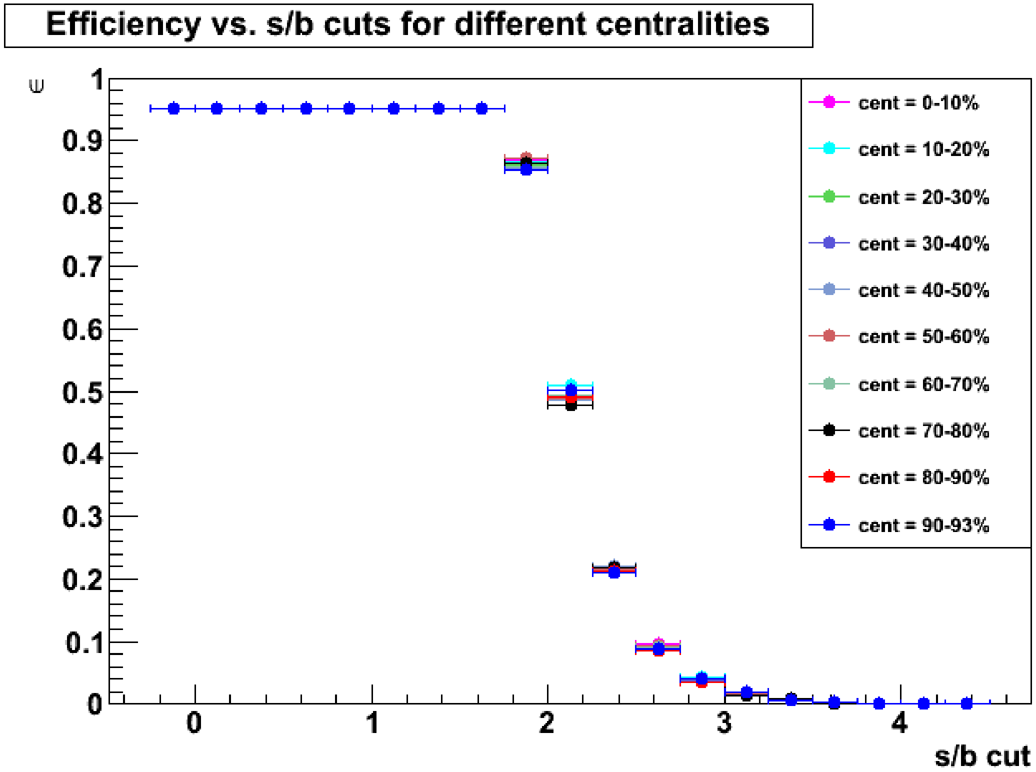
  - M (Poisson RV mean) and tau (Exp. RV decay const.) are hand tuned to match the RD pad charge distribution
    - Ten centrality bins of 10%
    - The long tail in RD is hard to reproduce (probably coming from jets? If so maybe can be added with some effort.)
    - This kind of detail matters for clusterizing
  - Using temporarily as a rough approximation to scintillation background





Optimizing s/b precluster selection cut

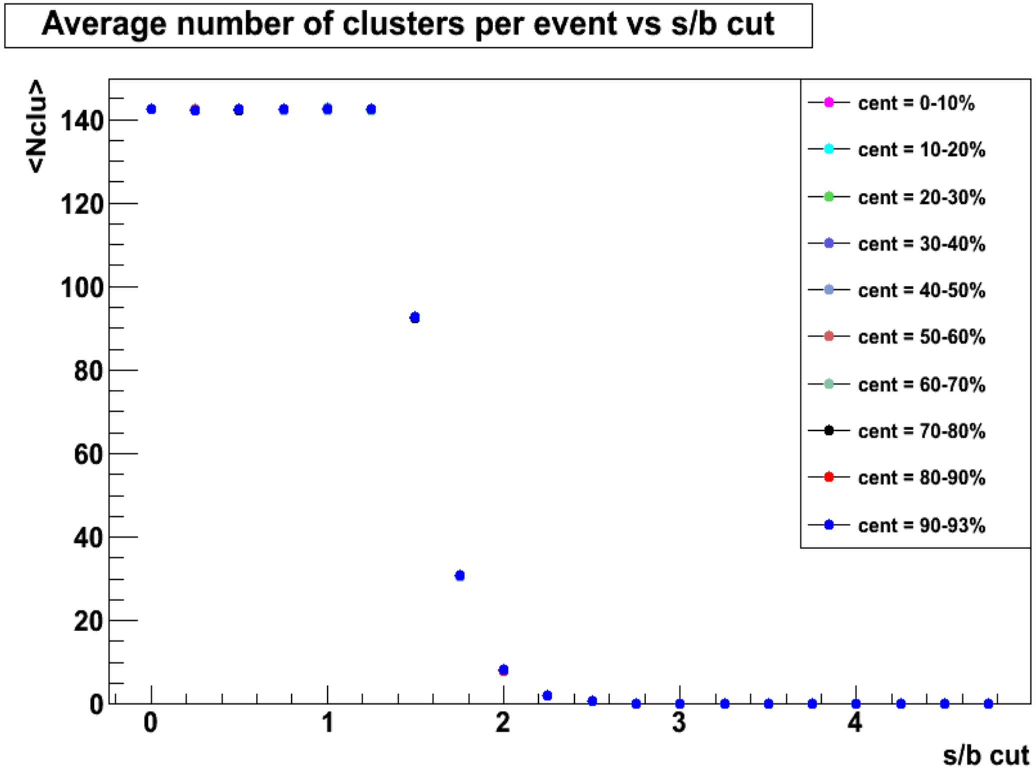
- Single electron cluster efficiency vs. s/b cut
  - Fraction of simulated single electrons that get associated with a cluster
  - vs. s/b cut using 'faux' scintillation background tuned to different centrality selections



  - The sudden drop in efficiency happens at the same position for all centralities, which points to a possible problem with the background emulator
  - Accumulator or embedding should give a better picture

Average number of clusters per event

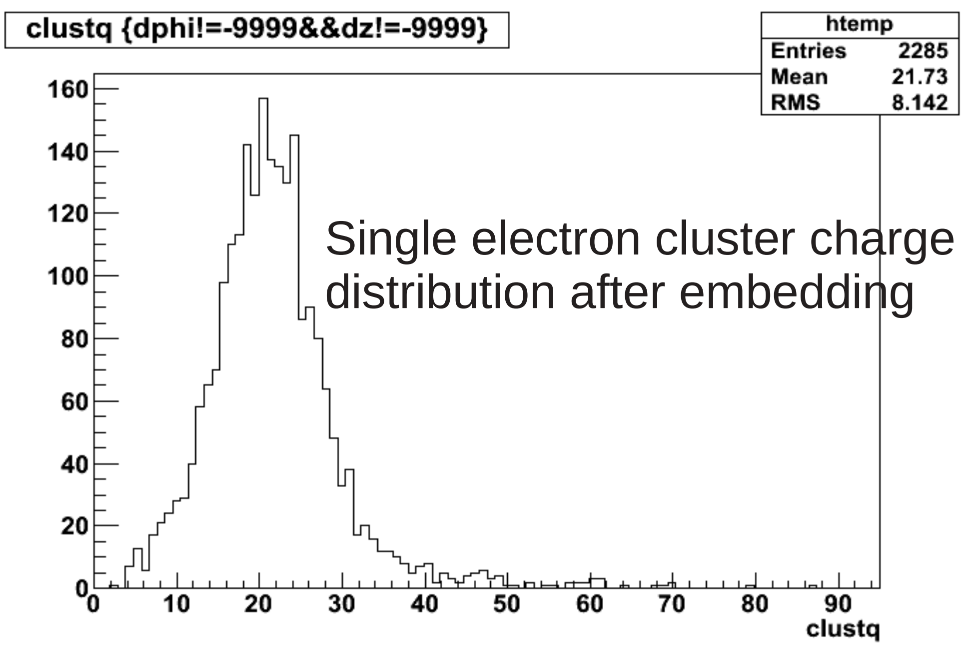
- Similar structure as for the efficiency
  - The drop in number of clusters occurs earlier than for the efficiency
  - With a s/b cut at 2, the efficiency is still > 90% but <Nclu> is down to less than 10



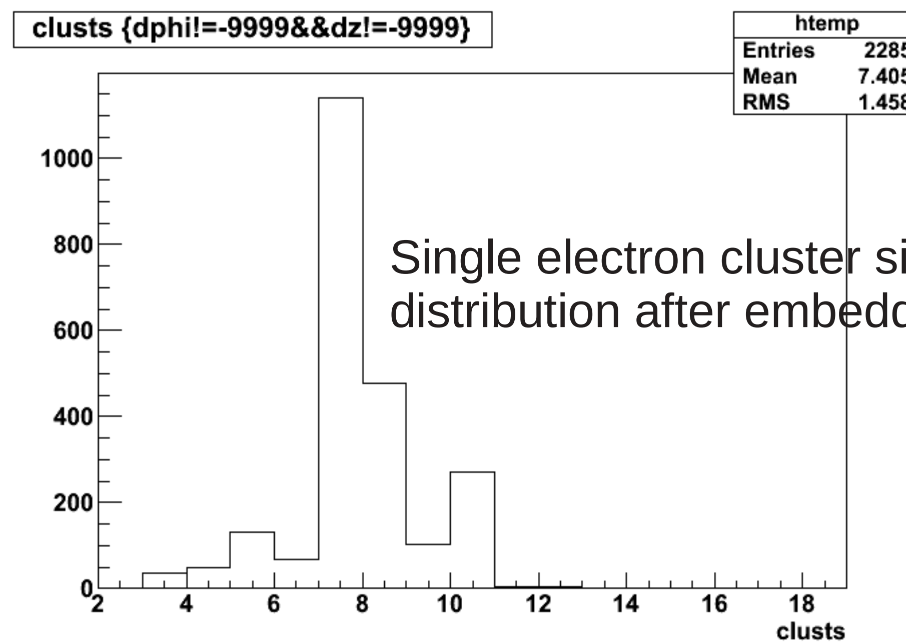
- This has to be confirmed by more realistic background env.
  - Embedding the simulated electron into events where there is no identified electron

Embedding MC Cerenkov response in RD

- Embedding is another option to see the effectiveness of a clusterization algorithm with real background
  - Simulate single (or double) electrons
  - Pad by pad add the signal from events in real data to the Cerenkov response from simulation
  - Run the clusterizer on merged HbdCellList
- Easy to implement real data event selection based on any criteria (bbcz, presence of electrons etc..) but not implemented yet. More to come...
- Embedding can be a useful tool to study the performance of a clusterizer (efficiency and stability in high background environment



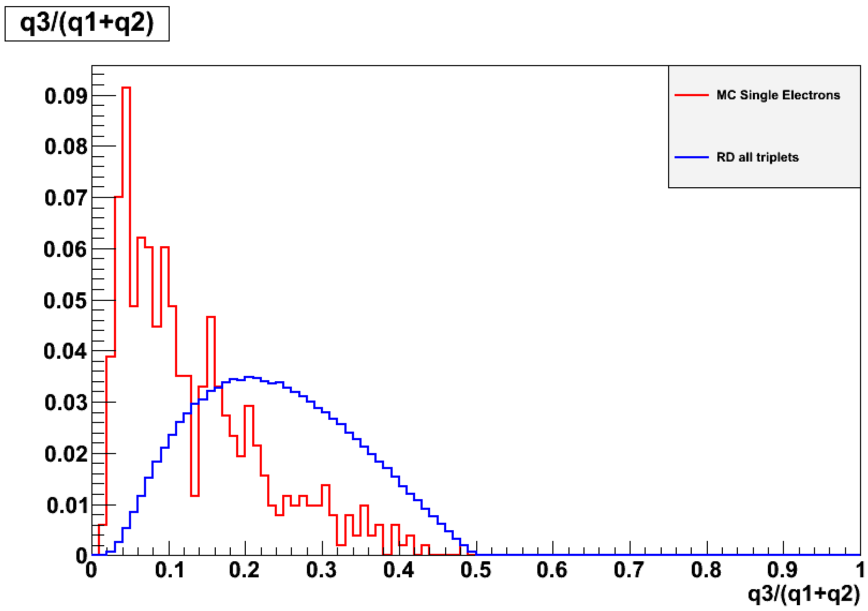
Single electron cluster charge distribution after embedding

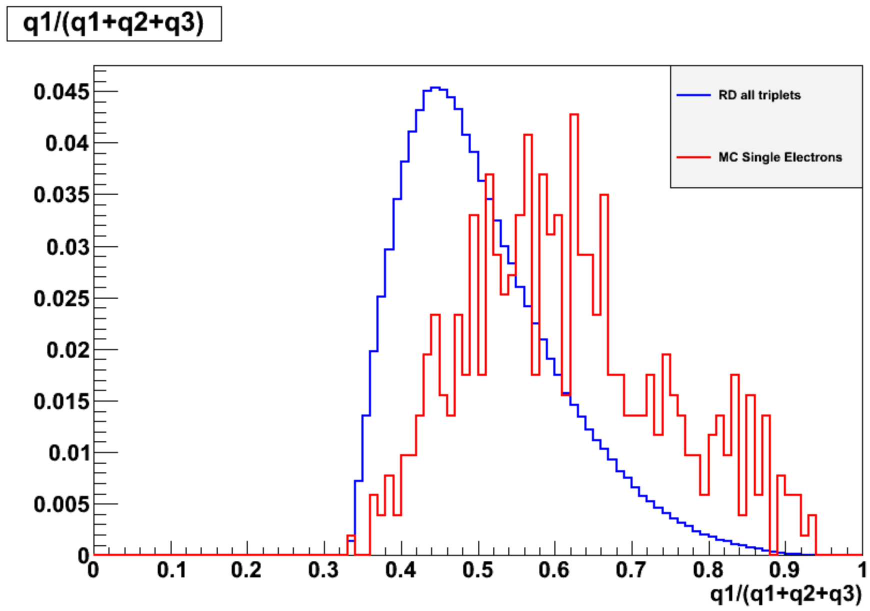


Single electron cluster size distribution after embedding

Cluster shape

- Distribution of charge among triplet member pads can be used to select preclusters
  - Tried two variables q1/(q1+q2+q3) and q3/(q1+q2) where q1 to q3 are the charges measured in the three pads of the triplet in decreasing order
  - There seems to be some possibility to use these or similar variables but it requires serious validation of the MC response of the HBD





Summary

- A new clusterization algorithm
  - Preclusterization: all triplets, s, b, shape
  - Selection:
    - Tighter selection criteria at this step => Loss of efficiency but also more stable results in terms of cluster size and charge
    - Optimization is simple and possible
  - Merging is straight forward if selection is done well
- The geometry use inside the clusterizer is validated using single electron simulation
- How a selection criteria can be optimized is demonstrated using s/b cut and faux scintillation background
- Other potential selection parameters (shape) distribs shown.
- Things left to do:
  - Make the scintillation background more realistic
  - See the doubles responses
  - Less urgent but still important: Optimize the code itself, make it leak free